Report

Ram Ayyala

2022-05-03

Introduction

The Healthcare Cost and Utilization Project(HCUP) provides the opportunity for state data organizations, hospital associations, private data organizations and the federal government to help create a massive family of healthcare databases that can help data analysts to investigate health policy isues such as health services' cost and quality, medical practice patterns, healthcare programs, and treatment outcomes from local market levels all the way to the national market level. In 1998, HCUP started merging all of this data to create what is known as the National Impatient Sample (NIS) data. The NIS contains key patient information such as patient demographics, length of stay at the hospital, and disease classifications, transfer status, and most importantly, risk of mortality. **Using this valuable information, it becomes possible to build a machine learning model to predict patient mortality and even determine what key factors contribute to patient death while hospitalized.** This model could help potentially mitigate patient mortality and help create health policies to better patient life.

Dataset Information

For this model, we will be using the NIS data set from 2012, which contains 200,000 randomly selected patients from all over the US. The data itself is an aggregation of discharge records from all participating HCUP hospitals. In order to predict our outcome of interest, which is patient mortality during hospitalization or the **DIED** feature, we will use the following available features: **AGE:**Age in years at admission coded 0-124 years **APRDRG_Risk_Mortality:**All Patient Refined DRG: Risk of Mortality Subclass: (0) No class specified (1) Minor likelihood of dying (2) Moderate likelihood of dying (3) Major likelihood of dying (4) Extreme likelihood of dying

APRDRG_Severity: All Patient Refined DRG: Severity of Illness Subclass: (0) No class specified (1) Minor loss of function (includes cases with no comorbidity or complications) (2) Moderate loss of function (3) Major loss of function (4) Extreme loss of function
CM_AIDS:AHRQ comorbidity measure: Acquired immune deficiency syndrome
CM_ALCOHOL:AHRQ comorbidity measure: Alcohol abuse
CM_ANEMDEF:AHRQ comorbidity measure: Deficiency anemias
CM_ARTH:AHRQ comorbidity measure: Rheumatoid arthritis/collagen vascular diseases
CM_BLDLOSS:AHRQ comorbidity measure: Chronic blood loss anemia
CM_COAG:AHRQ comorbidity measure: Coagulopathy
CM_DEPRESS:AHRQ comorbidity measure: Depression
CM_DM:AHRQ comorbidity measure: Diabetes, uncomplicated
CM_DMCX:AHRQ comorbidity measure: Diabetes with chronic complications
CM_DRUG:AHRQ comorbidity measure: Drug abuse
CM_HYPOTHY: AHRQ comorbidity measure: Hypothyroidism
CM_LIVER:AHRQ comorbidity measure: Hypothyroidism
CM_LYMPH:AHRQ comorbidity measure: Lymphoma
CM_METS:AHRQ comorbidity measure: Metastatic cancer

CM_OBESE:AHRQ comorbidity measure: Obesity
CM_PARA: AHRQ comorbidity measure: Paralysis
CM_PERIVASC:AHRQ comorbidity measure: Peripheral vascular disorders
CM_PSYCH:AHRQ comorbidity measure: Psychoses
CM_PULMCIRC:AHRQ comorbidity measure: Pulmonary circulation disorders
CM_RENLFAIL:AHRQ comorbidity measure: Renal failure
CM_TUMOR:AHRQ comorbidity measure: Solid tumor without metastasis
CM_VALVE: AHRQ comorbidity measure: Valvular disease
CM_WGHTLOSS:AHRQ comorbidity measure: Weight loss
DIED: Indicates in-hospital death: (0) did not die during hospitalization (1) died during hospitalization
FEMALE: Indicates gender for NIS beginning in 1998: (0) male (1) female
HOSP_DIVISION: Census Division of hospital (STRATA): (1) New England (2) Middle Atlantic (3) East North Central (4) West North Central (5) South Atlantic (6) East South Central (7) West South Central (8) Mountain (9) Pacific
LOS: Length of stay, edited
NCHRONIC: Number of chronic conditions
NDX: Number of diagnoses coded on the original record
NEOMAT: Assigned from diagnoses and procedure codes: (0) not maternal or neonatal (1) maternal diagnosis or procedure (2) neonatal diagnosis **ORPROC:**Major operating room procedure indicator: (0) no major operating room procedure (1) major operating room procedure
PAY1: Expected primary payer, uniform: (1) Medicare (2) Medicaid (3) private,including HMO (4) self-pay (5) no charge (6) other
RACE: Race, uniform coding: (1) white (2) black (3) Hispanic (4) Asian or Pacific Islander (5) Native American (6) other
TRAN_IN: Transfer in Indicator: (0) not a transfer (1) transferred in from a different acute care hospital [ATYPE NE 4 & (ASOURCE=2 or POO=4)] (2) transferred in from another type of health facility [ATYPE NE 4 & (ASOURCE=3 or POO=5,6)]

TRAN_OUT: Transfer out Indicator: (0) not a transfer (1) transferred out to a different acute care hospital (2) transferred out to another type of health facility
YEAR: Discharge year
ZIPINC_QRTL: Median household income national quartiles for patient's ZIP Code

Using these features, it is possible to build a model that can better patient care and pinpoint what could cause an increased patient mortality during hospitalization and potentially prevent patient death.

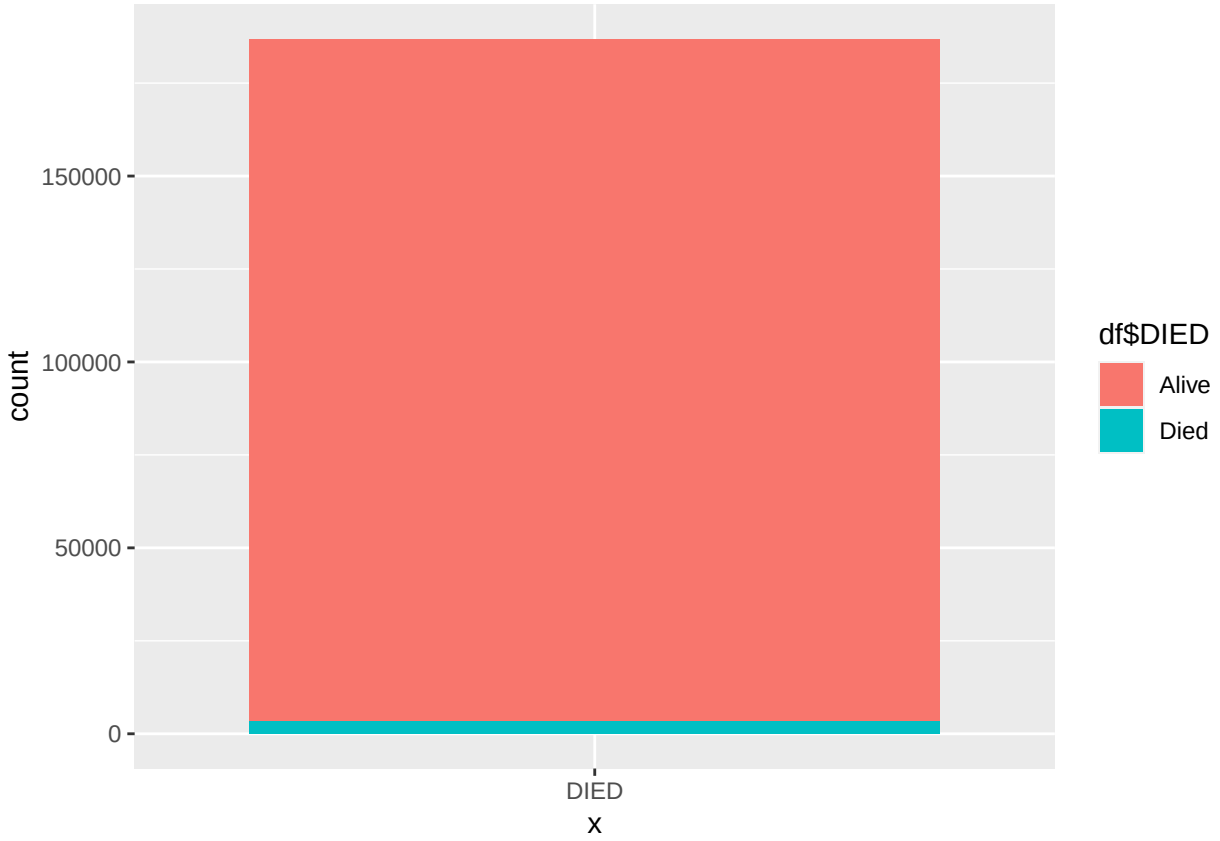
Methods

Data Cleaning

After data was loaded, relevant columns that would be deemed relevant to the model were selected from the original 175 features, reducing the number of features to 40. Then the data set itself was cleaned by examining features to determine if there were any odd values in the features such as were present in features like **DIED**, **FEMALE**, and **RACE**. From here, each feature was investigated to determine whether they could stay as their current type, or be converted into either numeric, factor, character, or integer. Finally, all NA and missing values were removed from the data set reducing the number of samples from 200,000 samples to 186861. From the data cleaning process and as shown in the bar plot below, it became evident that there was a large imbalance between the **majority class (Alive)** and the **minority class (Died)**, which meant that all scoring measures must be calculated using the **Area Under the Curve** metric as classification error can be heavily skewed with such large imbalances.

```
ggplot(df, aes(x="DIED",fill=df$DIED)) + geom_bar()
```

```
## Warning: Use of ‘df$DIED‘ is discouraged. Use ‘DIED‘ instead.
```



Modeling

For the modeling part of this analysis, I chose 3 different models: **1) Logistic Regression 2) Random Forest 3) Lasso Regression**

Logistic Regression

The Logistic Regression model was chosen because it is the baseline binary classifier in machine learning. Given that the outcome variable, **DIED**, is a binary variable, the use case of the Logistic Regression Model is thus justified. For the model, forward selection was employed to determine which features of the selected features should be used to give the best logistic regression model. The forward selection feature analysis found that the only relevant feature was the **APRDRG_Risk_Mortality** feature, which essentially is the patient mortality risk. Because this was the only feature deemed significant, I chose to include the following patient demographic features: **1) AGE 2) RACE 3) LOS 4) FEMALE** These features were included as with only one feature, the logistic regression model was at risk of having a high bias. Once feature selection was completed, hyper parameter tuning was employed on the **epsilon** and the **maxit** parameters of the Logistic Regression Model to determine the optimal parameters. In order to further mitigate the high bias risk factor, I employed cross validation (cv=5) on both the feature selection and the hyper parameter tuning process. AUC was employed as the scoring metric due to the imbalanced classes in the target outcome variable.

Random Forest

The Random Forest Model was chosen due to its high usage in binary classification problems and its ability to tune the parameters of the model accordingly to the importance of the features. In this model, all features were used due to this capability. In addition to Random Forest, a bagging Random forest model was employed to reduce the risk of over fitting. AUC was employed as the scoring metric due to the imbalanced classes in the target outcome variable.

Lasso Regression

The Lasso Regression Model was chosen over other glmnet models such as RIDGE Regression, as they use all of the features, which would increase the risk of an over fitted model. Furthermore, as was shown in the Logistic Regression model, not all features were significant to predicting patient mortality, furthering the justification of the Lasso Regression Model Use. Due to its inherent need to minimize the cost function, the Lasso Regression model will automatically select features that is deems significant, thus preventing the need to use a feature selection algorithm. In order to further mitigate the high bias risk factor, I employed cross validation (cv=5) on the resampling method. AUC was employed as the scoring metric due to the imbalanced classes in the target outcome variable.

Once all models were created, the AUC scores from all models were calculated to compare performance to choose the best model.

Results

Logistic Regression:

```
nis.at$tuning_result %>% knitr::kable(caption="Optimal Parameters for Logistic Regression")
```

Table 1: Optimal Parameters for Logistic Regression

epsilon	maxit	learner_param_vals	x_domain	classif.auc
0	50	1e-08, 5e+01	1e-08, 5e+01	0.9340465

From the hyper parameter tuning, the optimal parameters were found to be **epsilon=0** and a **maxit=50**.

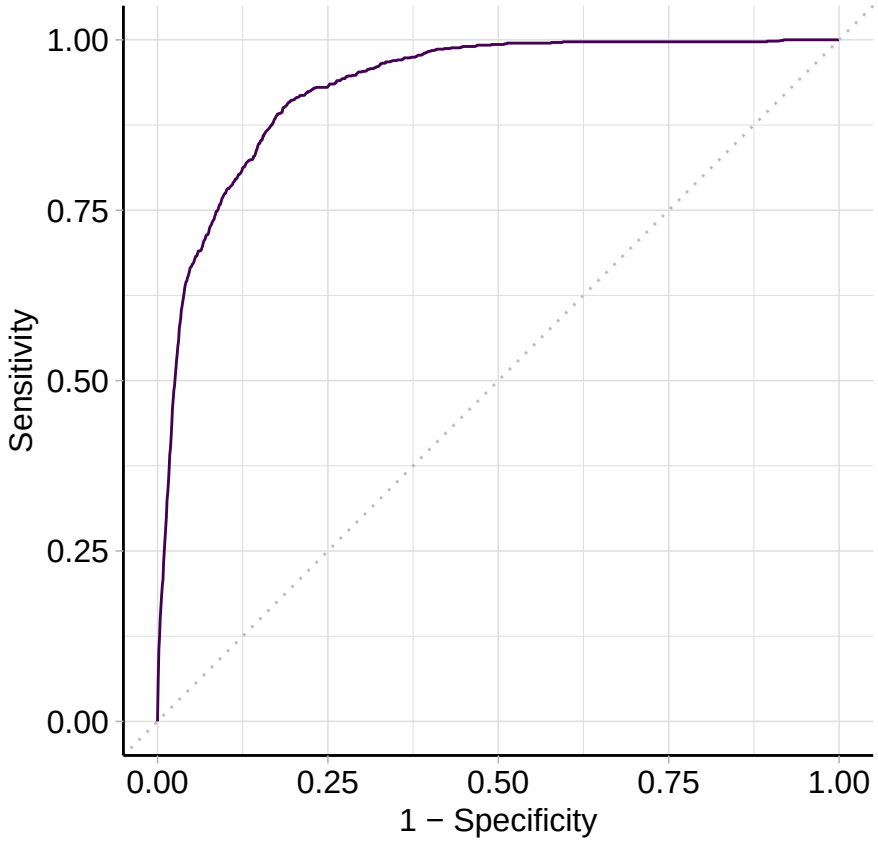
```
data.frame(model=c("Training","Test"), AUC_Scores=c(nis.at$tuning_result$classif.auc,logreg_pred$score(
```

Table 2: Training and Test AUC for Logistic Regression

	model	AUC_Scores
	Training	0.9340465
classif.auc	Test	0.9309397

The Training AUC Score was 0.9340465 while the Test AUC Score was 0.9309397.

```
autoplot(logreg_pred, type="roc")
```



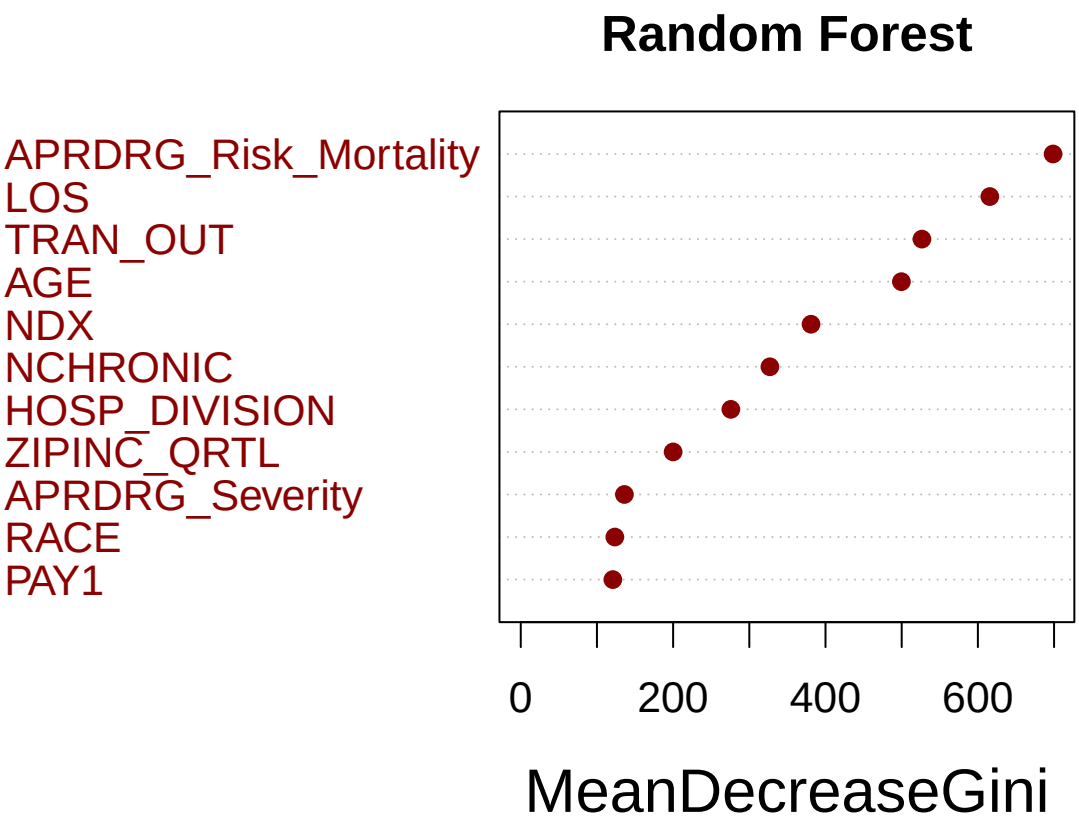
The ROC Curve for the Test Score is shown above.

Random Forest:

Random Forest Alone:

The Random Forest model had an error rate of 7.4760514.

```
varImpPlot(rf, cex.lab=1.5, cex.axis=2, cex=1.3,
           n.var=ncol(df)-29, main="Random Forest", pch=16, col='red4')
```



Above is the Variable Importance Plot. Like the Logistic Regression Model, the Random Forest Model deemed the APRDRG_Risk_Mortality Feature as the most important variable. Only features with a score of above 100 were selected for the plot.

```
importance(rf)[order(importance(rf)[,1], decreasing = TRUE),]%>% knitr::kable(caption="Variable Importa
```

Table 3: Variable Importance Scores for Random Forest

	x
APRDRG_Risk_Mortality	698.672432
LOS	615.669329
TRAN_OUT	526.568868
AGE	499.614388
NDX	380.996407
NCHRONIC	327.021889
HOSP_DIVISION	275.784306
ZIPINC_QRTL	199.980612
APRDRG_Severity	136.057362
RACE	123.494643
PAY1	120.861418
TRAN_IN	77.290925
FEMALE	54.423888
CM_DM	45.613673
CM_RENLFAIL	44.705650
CM_ANEMDEF	43.154132

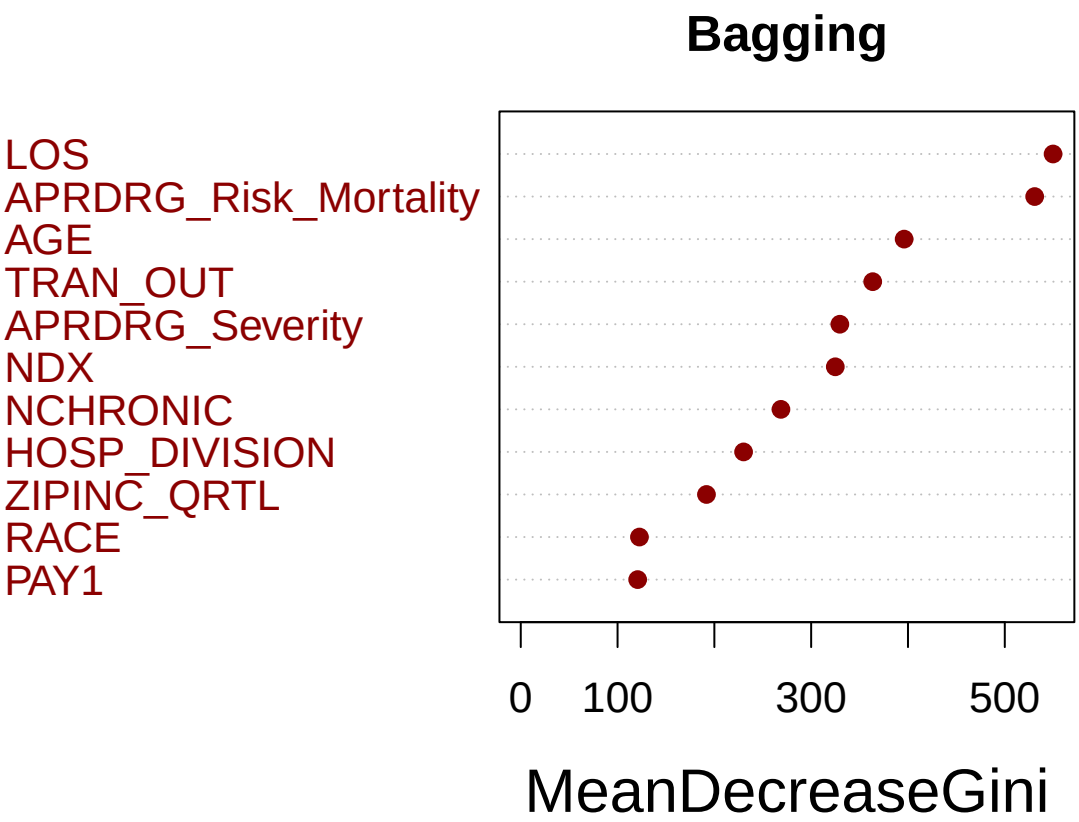
	x
ORPROC	38.081655
CM_WGHTLOSS	36.757857
CM_HYPOTHY	35.836717
CM_COAG	32.375000
CM_DEPRESS	30.590759
CM_METS	28.711030
CM_PERIVASC	28.453430
CM_OBESE	28.005837
CM_PULMCIRC	25.684997
CM_VALVE	24.794527
CM_LIVER	21.538326
CM_ALCOHOL	20.126646
CM_TUMOR	19.940231
CM_PARA	19.839317
CM_PSYCH	18.931255
CM_DMCX	18.618937
CM_LYMPH	15.328882
CM_ARTH	15.199643
CM_DRUG	11.931060
CM_BLDLOSS	10.507049
NEOMAT	4.647565
CM_AIDS	2.160337
YEAR	0.000000

Above is the Importance Variable Table with the full list of variables and there importance scores. As shown in the Variable Importance plot, the APRDRG_Risk_Mortality feature has the highest score with a value of 698.6724315. Notably, the APRDRG_Severity was not the second highest while the LOS or length of stay feature was ranked with a score of 615.6693293.

Random Forest with Bagging

The Random Forest Bagging model had an error rate of `sum(rf_bg$err.rate[,1])`.

```
varImpPlot(rf_bg, cex.lab=1.5, cex.axis=2, cex=1.3,
           n.var=ncol(df)-29, main="Bagging", pch=16, col='red4')
```



Above is the Variable Importance Plot. Unlike the Logistic Regression Model and the Random Forest Model, the Random Forest Bagging Model deemed the LOS or length of stay Feature as the most important variable. Only features with a score of above 100 were selected for the plot.

```
importance(rf_bg)[order(importance(rf_bg)[,1], decreasing = TRUE),] %>% knitr::kable(caption="Variable
```

Table 4: Variable Importance Scores for Random Forest Bagging

	x
LOS	549.879114
APRDRG_Risk_Mortality	530.885689
AGE	396.043443
TRAN_OUT	363.585811
APRDRG_Severity	329.626250
NDX	324.926247
NCHRONIC	268.767450
HOSP_DIVISION	230.197082
ZIPINC_QRTL	191.841152
RACE	122.622827
PAY1	120.771643
TRAN_IN	77.656159
FEMALE	71.420941
CM_ANEMDEF	57.394877
CM_DM	54.146200
CM_RENLFAIL	52.122566

	x
ORPROC	46.943915
CM_HYPOTHY	44.051088
CM_COAG	41.411657
CM_WGHTLOSS	41.215714
CM_DEPRESS	35.362865
CM_PERIVASC	35.165283
CM_OBESE	31.603226
CM_METS	30.604611
CM_PULMCIRC	29.721377
CM_VALVE	29.242293
CM_LIVER	25.583903
CM_ALCOHOL	23.888624
CM_PARA	23.789526
CM_TUMOR	22.301494
CM_DMCX	21.792029
CM_PSYCH	21.441735
CM_ARTH	15.130538
CM_LYMPH	15.037794
CM_DRUG	13.650011
CM_BLDLOSS	9.564134
NEOMAT	8.503024
CM_AIDS	2.568504
YEAR	0.000000

Above is the Importance Variable Table with the full list of variables and there importance scores. As shown in the Variable Importance plot, the LOS feature has the highest score with a value of 698.6724315. Again, the APRDRG_Severity feature was not the second highest while the APRDRG_Risk_Mortality feature was ranked with a score of 615.6693293.

```
rf_predict = predict(rf, newdata = df[test, ],
                                type='response')

roc_test = roc(df$DIED[test], as.numeric(rf_predict)-1)
```

```
## Setting levels: control = Alive, case = Died
```

```
## Setting direction: controls < cases
```

```
auc(roc_test)
```

```
## Area under the curve: 0.6856
```

The Test AUC Score for the Random Forest Model is shown above.

```
rf_bg_predict = predict(rf_bg, newdata = df[test, ],
                                type='response')

roc_test = roc(df$DIED[test], as.numeric(rf_bg_predict)-1)
```

```
## Setting levels: control = Alive, case = Died
```

```
## Setting direction: controls < cases
```

```
auc(roc_test)
```

```
## Area under the curve: 0.6419
```

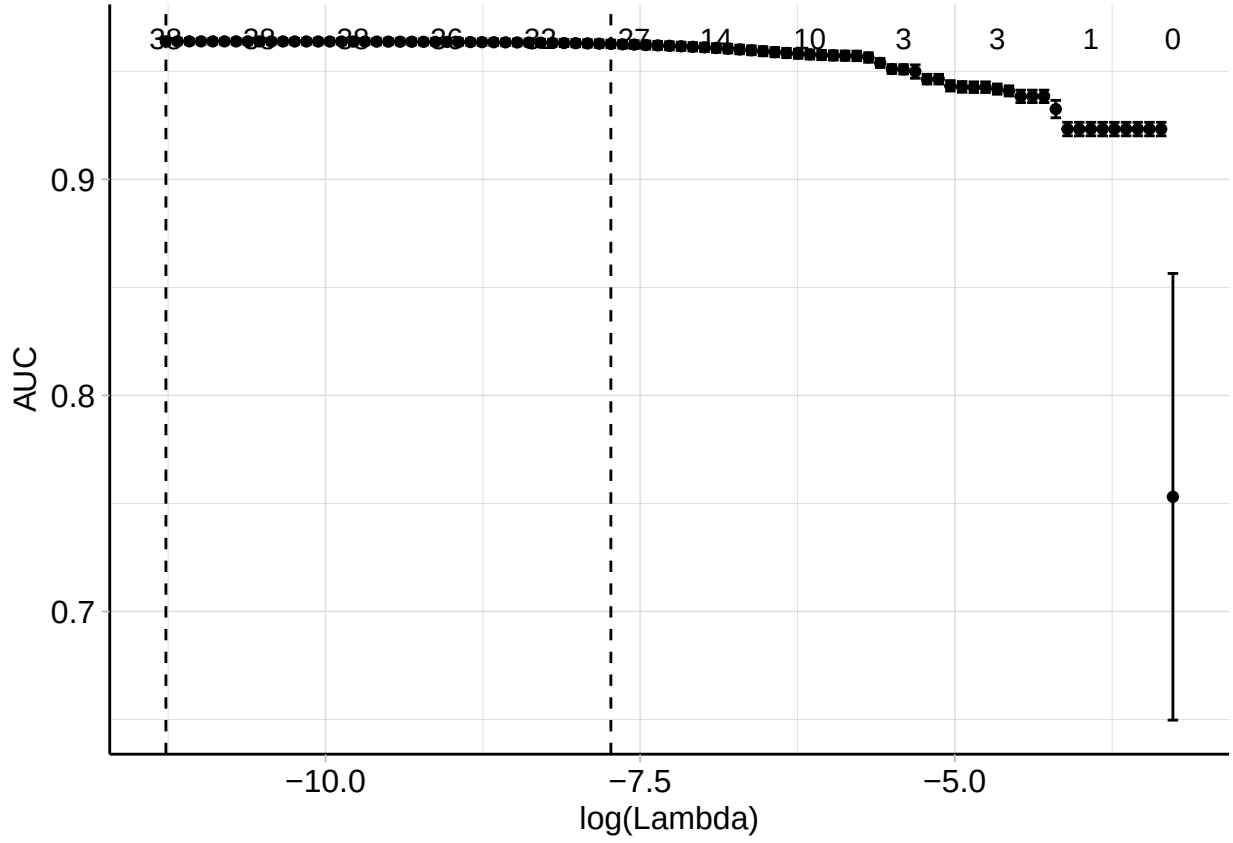
The Test AUC Score for the Random Forest Model is shown above.

LASSO Regression:

The Training AUC Score for the Lasso Regression Model was 0.9638846 while the Test AUC Score was 0.959704.

```
autoplot(lasso.lrn, type="roc")
```

```
## Warning: Ignoring unknown parameters: type
## Ignoring unknown parameters: type
```



The ROC Curve for the Test Score of the Random Forest Bagging Model is shown above.

Conclusion

From the models above, the best model by far was the Lasso Regression Model as it had the highest training and test AUC out of the three models proposed. The worst performing model by far was the Random

Forest Bagging Model which is probably due to the imbalanced data set. Most of these models treated the patient's mortality risk or the APRDRG_Risk_Mortality, as the priority feature to predict patient mortality, while including variables like the APRDRG_Severity, Length of Stay, and Race, and Age. The Logistic Regression Model, while being second in terms of model performance, was probably affected by the forward feature selection which reduced our number of features to a singular feature. While I did train and test the model using the mortality risk and demographics, this doesn't change the fact that the model itself was limited in terms of the number of features examined. The Random Forest models performed very poorly due to the imbalanced data sets. Even with Bagging added to the model, the performance was still poor, even dropping performance below the standard Random Forest Model. The Lasso Regression Model probably worked the best due to its ability to select features on its own and determine their significance to the targeted outcome. In the future, it would be best to work with a balanced data set or even balance this current data set via methods like oversampling, under-sampling or even using the SMOTE algorithm to account for imbalance. Moreover, it would be interesting to explore the other features left out from this study and see if any other feature would be a better predictor than than the patient mortality risk feature. Overall, using the Lasso Regression Model is probably the best method to predict patient mortality.